AF/2825

Docket No.: 64965-126 (formerly 52352-317) **PATENT**

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of	:	**RESPONSE UNDER 37 CFR 1.116**
	:	**EXPEDITED PROCEDURE**
	:	
Jacques WONG, et al.	:	
	:	
Serial No.: 09/517,518	:	Group Art Unit: 2825
	:	
Filed: March 02, 2000	:	Examiner: Annette M. THOMPSON
	:	

Corres. and Mail
BOX AF

For: BOTTOM-UP APPROACH FOR SYNTHESIS OF REGISTER TRANSFER LEVEL (RTL) BASED DESIGN (As Amended)

TRANSMITTAL OF APPEAL BRIEF

Assistant Commissioner for Patents
Washington, DC 20231

Sir:

Submitted herewith in triplicate is Appellants' Appeal Brief in support of the Notice of Appeal filed August 20, 2002. Please charge the Appeal Brief fee of $320.00 to Deposit Account 500417.

To the extent necessary, a petition for an extension of time under 37 C.F.R. 1.136 is hereby made. Please charge any shortage in fees due in connection with the filing of this paper, including extension of time fees, to Deposit Account 500417 and please credit any excess fees to such deposit account.

Respectfully submitted,

MCDERMOTT, WILL & EMERY

B. Y. Mathis
Registration No. 44,907

600 13th Street, N.W.
Washington, DC 20005-3096
(202) 756-8000 BYM/GZR:kap
Date: September 11, 2002
Facsimile: (202) 756-8087

TABLE OF CONTENTS

#15 Appeal Brief
9/17/02

Docket No.: 64965-126 (formerly 52352-317) **PATENT**

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of	:	**RESPONSE UNDER 37 CFR 1.116**
	:	**EXPEDITED PROCEDURE**
Jacques WONG, et al.	:	
Serial No.: 09/517,518	:	Group Art Unit: 2825
Filed: March 02, 2000	:	Examiner: Annette M. THOMPSON

For: BOTTOM-UP APPROACH FOR SYNTHESIS OF REGISTER TRANSFER LEVEL (RTL) BASED DESIGN (As Amended)

APPEAL BRIEF

Assistant Commissioner for Patents
Washington, DC 20231

Sir:

This Appeal Brief is submitted in support of the Notice of Appeal filed August 20, 2002.

I. REAL PARTY IN INTEREST

Advanced Micro Devices, Inc. is the real party in interest in the pending application.

II. RELATED APPEALS AND INTERFERENCES

Appellants are unaware of any related Appeals and Interferences.

III. STATUS OF CLAIMS

Claims 1-3 and 5-17 are pending in this application, of which claims 1-3 and 5-17 have been finally rejected. It is from this final rejection that this Appeal is taken.

IV. STATUS OF AMENDMENTS

Subsequent to the imposition of the final Office Action dated May 20, 2002 (hereinafter "the Office Action"), an Amendment was filed on July 22, 2002 to obviate an objection made to the Title of the Invention as well as a potential objection to the Abstract. The Amendment also amends claim 10 to obviate a minor informality raised in the Office Action. It is assumed that these amendments were entered.

V. SUMMARY OF INVENTION

The present invention relates to the field of integrated circuit design, and more particularly, to an approach for synthesizing Register Transfer Logic (RTL) based designs. See, page 1, lines 6-7 of the specification, for example.

Synthesizing RTL based designs is a process of translating a hardware descriptive language file that embodies the desired logical functionality of a hardware design into a number of interconnected logic gates that should not only provide the desired logical functionality, but also provide desired performance, e.g., the logic gates should meet a designer's timing constraints. See, page 1, lines 9-15 of the specification, for example.

One approach to synthesizing RTL designs is known as the "top-down" synthesis approach, where timing constraints are imposed on integrated circuit devices as a whole, i.e., at the pin level of an integrated circuit. Unfortunately, as logic designs grow increasingly larger and more complex, the top-down synthesis approach tends to create timing bottlenecks, which in turn can slow the synthesis process of a design thus increasing costs and time to market. See, page 1, lines 16-20 of the specification, for example.

The present invention works to alleviate the above-mentioned problems by using a hybrid of top-down and bottom-up approaches. For example, a register transfer level based design of a particular top-level system can be conceptually broken down into a number of sub-modules, and individual time budgets for each sub-module can then be determined based on timing requirements of the top-level system. Then for each sub-module, gate-level designs of the sub-modules can be (1) **synthesized** based on the determined time budgets, then (2) **tested** for conformance with individual sub-module gate-level design requirements. As the various sub-modules are synthesized and validated, they are integrated to form a top level gate design. The top-level design, in turn, can be tested for conformance with top-level design requirements and a top-level netlist can be generated. See, claims 1 and 10, for example.

VI. ISSUE

Whether claims 1-3 and 5-17 are anticipated by Dupenloup under 35 U.S.C. § 102(e).

VII. GROUPING OF CLAIMS

Claims 1 stands or falls alone; claims 2-3 and stand or fall together as a group; claim 4 stands or falls alone; claims 5 and 13 stand or fall together as a group; claims 6-8 stand or fall together as a group; claims 9 and 17 stand or fall together; claims 10-11 stand or fall together as a group; claim 12 stands or falls alone; and claims 14-16 stand or fall together as a group.

VIII. THE ARGUMENT

It is submitted that the rejection of claims 1-3 and 5-17 by the Examiner's rejection is predicated upon numerous inaccurate factual determinations. A rejection for lack of novelty under 35

USC §102 requires the identical disclosure in a single reference of each element of a claimed invention. *Helfix, Ltd. v. Loc-Bloc, Ltd. 54 USPQ2d 1299 (Fed. Cir. 2000); TD Corporation v. Lydall, Inc. 159 F.3d. 534, 48 USPQ2d 1321 (Fed. Cir. 1998); Electro Medical Systems S.A. v. Coopoer Life Science, Inc., 34 F.3d. 1048, 32 USPQ2d 1017 (Fed. Cir. 1994).*

The Dupenloup Reference

Dupenloup discloses a method for generating scripts to synthesize RTL code. See, Abstract, for example. The Dupenloup method uses a complementary approach of top-down characterization and bottom-up synthesis steps. As shown in figure 19, the top-down step is used to provide constraints, time budgets and other information to be met by each module of a larger design. See also, col. 43, lines 16-19, for example. Once the constraints are formed, the various modules are synthesized, then integrated along a predetermined design hierarchy. See, col. 41, lines 1-12, for example.

A. **The rejection of claims 1 and 12 under 35 U.S.C. §102(e) is improper**

Dupenloup does not teach or suggest a method of synthesizing a register transfer level based design of a system including at least the steps of determining a plurality of sub-modules of a top level system, determining individual time budgets for each sub-module based on timing requirements of the top-level system, **synthesizing** gate-level designs of the sub-modules based on the determined time budgets for the individual sub-modules, and **testing** the gate-level designs for conformance with gate-level design requirements of the individual sub-modules, then integrating the gate-level designs of the individual sub-modules to form a top level design, as recited in independent claim 1 and similarly recited in dependent claim 12.

While Dupenloup discloses **synthesizing** various modules based on time-constraints, nowhere

does Dupenloup disclose, suggest or even appreciate any form of individual sub-module **testing**. See, col. 43, lines 8-26, for example. Although the December 7, 2001 non-final Office Action asserts that Dupenloup discloses "testing the gate-level designs . . . prior to integrating the gate-level designs to form the top-level design (see, page 5, paragraph 7.4 of the Office Action), nowhere do the cited passages disclose performing any testing on individual sub-modules.

The Examiner's position is based upon the following text at col. 43, lines 8-26:

> Referring to FIG. 19, the interactive improvement process begins with initial mapping utilizing bottom-up synthesis technique 450 with each module being assigned default constraints, time budgets, and clock definitions. After the initial mapping is completed, top-down characterization 451 is performed. **Top-down characterization provides constraints, time budgets, and other information required to be met by each of the modules.** Then, the **constraints** determined by the characterization step are used to **resynthesize** each of the modules using bottom-up **resynthesis** technique 452. The top-down characterization step 451 and bottom-up **resynthesis** 452 steps are iterated 455 until all **constraints** are met by each of the modules being **synthesized** and gate count for each of the modules are stable. {bolded emphasis added}

As shown by the above text, the process described by Dupenloup shows that modules can be synthesized (resynthesized) based on top-level constraints, but there is no disclosure that individual modules are **tested** once synthesized. In fact, the term "test" or its derivatives or equivalents is nowhere to be found in the Examiner's cited passages.

In response to Applicants' response to the non-final Office Action, the Examiner later asserted in the subsequent final Office Action that Dupenloup tests gate-level designs for conformance with gate-level design requirements citing col. 30, lines 12-57 and Fig. 1. See, page 4, second paragraph. However, **these cited passages do not refer to testing of individual sub-modules.**

To the contrary, while Fig. 1 does refer to a gate-level verification step **110**, Fig. 1 is an example where gate-level verification occurs **after** a synthesis step **106** is completed, i.e., not during the synthesis step **106**. In contrast, the invention of claim 1 requires testing gate level designs of

individual sub-modules before integrating such gate-level designs.

Further, a review of Fig. 14, which depicts a three-step process for **bottom-up synthesis** used by Dupenloup, shows module synthesis and integration, **but again does not depict any intermediate testing of sub-modules using gate-level design requirements of individual sub-modules**. See also, col. 41, lines 1-12.

Still further, nowhere does Dupenloup show where any gate-level requirements are even generated for individual modules, much less indicate where any testing of gate-level designs for conformance with gate-level design requirements of individual sub-modules is performed. While the Office Action asserts that "determining individual time budgets for each sub-module" is supported by col. 42, lines 29-31, col. 16, lines 6-8, col. 43, line 20 and col. 43, lines 15-18, these passages are either (1) directed to **using time budgets in general and do not even mention sub-modules,** or (2) directed to sub-modules **without disclosing or suggesting time budgets for sub-modules**. Accordingly these passages cannot support the Office Action's assertion as the passages are either not relevant or do not teach or suggest testing gate level designs of sub-modules before integration.

Applicants also wish to draw attention to the May 20, 2002 Final Office Action that states that Applicants asserted that "Dupenloup does not use the word 'test' and therefore Dupenloup cannot anticipate Applicants' limitation". The Examiner misconstrues Applicants' assertion. A review of Applicants' March 7, 2002 response to the non-final Office Action shows that Applicants asserted that "nowhere in Dupenloup does the term 'test' **occur in conjunction with individual sub-modules**." **{bolded emphasis added}** To date, the Examiner fails to provide a single passage in Dupenloup that discloses where individual sub-modules are tested in any form.

Further, the Examiner has repeatedly ignored the Applicants' assertion regarding Dupenloup's failure to disclose **testing of individual sub-modules**. The Examiner's response of the final Office

Action is limited to "Dupenloup uses the word **verification** and verification and test are synonymous in the art of integrated circuits." {bolded emphasis added} While the Examiner is literally correct in that "Dupenloup uses the word verification", the Examiner fails to show where Dupenloup uses the word "verification" in the context of testing individual sub-modules. In fact, the term "verification" is used twice in Dupenloup; the first instance occurring on col. 2, line 59 of the specification, the second instance occurring on col. 13, line 54 of the specification. A review of these passages and any supporting text clearly shows that in neither instance does Dupenloup discuss "verification" in the context of individual sub-modules.

B. **The rejection of claims 2-3 and 10-11 under 35 U.S.C. §102(e) is improper**

Dupenloup does not teach or suggest a method of synthesizing an RTL-based design of a system including generating gate-level netlists for the gate-level designs of each of the sub-modules, and integrating the gate-level designs of the individual sub-modules, as recited in independent claim 10 and dependent claim 2.

The Examiner asserts that Dupenloup suggests "generating gate-level netlists" in disclosing "progressive synthesis of sub-modules" at col. 49, line 51 et seq. See, page 4, paragraph 5 of the final Office Action. However, module synthesis and netlist generation for a module are two separate and distinctly different activities. Should any activity related to netlists be included regarding "progressive synthesis of sub-modules", then Dupenloup would have certainly stated such as Dupenloup is not indisposed to use the term "netlist", which in fact occurs at least 146 times in the Dupenloup specification. Applicants respectfully point to the Examiner's failure to provide a single example where Dupenloup uses the term "netlist" in the context of describing the generation of a netlist from a sub-module.

Instead, the Examiner relies on a passage of Dupenloup that discloses that a "dump script technique used to extract the generic netlist from a Design Compiler has several benefits" (see col. 21, lines 29-33). While the Examiner states that a "dump script technique used to extract the generic netlist from a Design Compiler has several benefits," the Examiner fails to show how this passage somehow suggests generating a netlist from a sub-module, via a "Design Compiler" or otherwise.

While the Examiner cites col. 1, lines 34-36 and Fig.1 to support its claim that Dupenloup discloses generating gate-level netlists for the gate-level designs of each of the sub-modules (see page 4, first paragraph of the final Office Action), the synthesis step **106** of Fig. 1, which merely indicates that some synthesis step occurs. The cited passage states that the "IC design, as expressed by the RTL code, is then synthesized to generate a gate-level description, or a netlist". Neither the cited figure nor passage teaches, suggests or even indicates any appreciation for generating gate-level netlists for individual sub-modules.

C. **The rejection of claims 5 and 13 under 35 U.S.C. §102(e) is improper**

Dupenloup does not teach or suggest a method of synthesizing an RTL-based design of a system wherein testing gate-level designs includes performing static timing analysis on individual sub-modules for conformance with timing requirements for individual sub-blocks, as recited in dependent claims 5 and 13.

The Examiner asserts that Dupenloup discloses performing static timing analysis on individual sub-modules at col. 11, line 55 to col. 12, line 42 and col. 13, lines 33-36. While the passages do refer to static timing analysis, i.e., defining clock domains, relationship between clocks, and interaction between clock domains, neither passage refers to performing static timing analysis on individual sub-modules. In fact, a review of the cited passages reveals that these passages do not disclose, suggest or

even appreciate performing static timing analysis on individual sub-modules.

D. **The rejection of claims 6-8 and 14-16 under 35 U.S.C. §102(e) is improper**

Dupenloup does not teach or suggest a method of synthesizing an RTL-based design of a system wherein the gate-level netlists are generated for the sub-modules **only if** the timing requirements for the individual sub-modules are met, as recited in dependent claims 6 and 14.

The Examiner asserts that Dupenloup discloses generating gate-level netlists for the sub-modules **only if** the timing requirements for the individual sub-modules are met at col. 71, lines 9-23. However, a review of this passage (as well as the following text and Fig. 27) reveals that there is no support for the Examiner's assertion. To the contrary, the cited passages do not even refer to gate-level netlist, much less generating gate-level netlist for individual sub-modules.

E. **The rejection of claims 9 and 17 under 35 U.S.C. §102(e) is improper**

Dupenloup does not teach or suggest a method of synthesizing an RTL-based design of a system having a step of verifying conformance of the gate-level designs that includes performing dynamic simulations on the gate-level designs (of the individual sub-modules), as recited in dependent claims 6 and 14.

The Examiner asserts that Dupenloup discloses verifying conformance of the gate-level designs that includes performing dynamic simulations on the gate-level designs of the individual sub-modules at col. 10, lines 33-51. While the cited passage is directed to testing and simulation, the cited passage does not refer to performing dynamic simulations on the gate-level designs of individual sub-modules. In fact, a review of the cited passage reveals that the passage does not disclose, suggest or even appreciate performing dynamic simulations on individual sub-modules.

F. **Conclusion:**

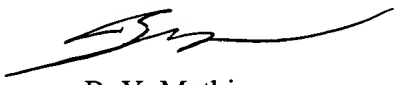
Based upon the arguments submitted ***supra***., it is respectfully submitted that the record does not establish that any of the appealed claims are anticipated under 35 U.S.C. §102. Appellants, therefore, respectfully solicits reversal of the imposed rejections under 35 U.S.C. §102 as erroneous.

IX. PRAYER FOR RELIEF

To the extent necessary, a petition for an extension of time under 37 C.F.R. 1.136 is hereby made. Please charge any shortage in fees due in connection with the filing of this paper, including extension of time fees, to Deposit Account 500417 and please credit any excess fees to such deposit account.

Respectfully submitted,

MCDERMOTT, WILL & EMERY



B. Y. Mathis
Registration No. 44,907

600 13th Street, N.W.
Washington, DC 20005-3096
(202) 756-8000 BYM/GZR:kap
Date: September 11, 2002
Facsimile: (202) 756-8087

X. **APPENDIX**

1. A method of synthesizing a register transfer level (RTL) based design of a system comprising the steps of:

determining a plurality of sub-modules of a top level system;

determining individual time budgets for each sub-module based on timing requirements of the top-level system;

synthesizing gate-level designs of the sub-modules based on the determined time budgets for the individual sub-modules;

testing the gate-level designs for conformance with gate-level design requirements of the individual sub-modules, then integrating the gate-level designs of the individual sub-modules to form a top level design;

testing the top-level design for conformance with top-level design requirements; and

generating a top-level netlist when the top-level design conforms to the top-level design requirements.

2. The method of claim 1, further comprising generating gate-level netlists for the gate-level designs of each of the sub-modules.

3. The method of claim 2, wherein the step of integrating the gate-level designs includes integrating the gate-level netlists of the sub-modules.

5. The method of claim 1, wherein testing the gate-level designs includes performing static timing analysis on the individual sub-modules for conformance with timing requirements for the individual sub-blocks.

6. The method of claim 5, wherein the gate-level netlists are generated for the sub-modules only if the timing requirements for the individual sub-modules are met.

7. The method of claim 6, wherein the step of synthesizing is re-performed and the gate-level designs are re-tested in an iterative manner to verify conformance of the gate-level designs with the timing requirements of the individual sub-modules.

8. The method of claim 7, wherein the step of synthesizing is further based on wire loads and input/output loads/drivers.

9. The method of claim 8, wherein the step of verifying conformance of the gate-level designs includes performing dynamic simulations on the gate-level designs.

10. A method of synthesizing a register transfer level (RTL) based design of a system comprising the steps of:

determining sub-modules of a top level system;

determining individual time budgets for each sub-module based on timing requirements of the top-level system

synthesizing gate-level designs of the sub-module based on the determined time budgets for the individual sub-modules;

integrating the gate-level designs of the individual sub-modules to form a top level design;

testing the top-level design for conformance with top-level design requirements;

generating gate-level netlists for the gate-level designs of each of the sub-modules; and

generating a top-level netlist when the top-level design conforms to the top-level design requirements.

11. The method of claim 10, wherein the step of integrating the gate-level designs includes integrating the gate-level netlists of the sub-modules.

12. The method of claim 11, further comprising testing the gate-level designs for conformance with gate-level design requirements of the individual sub-modules prior to integrating the gate-level designs to form the top-level design.

13. The method of claim 12, wherein testing the gate-level designs include performing static timing analysis on the individual sub-modules for conformance with timing requirements for the individual sub-blocks.

14. The method of claim 13, wherein the gate-level netlists are generated for the sub-modules only if the timing requirements for the individual sub-modules are met.

15. The method of claim 14, wherein the step of synthesizing is re-performed and the gate-level designs are re-tested in an iterative manner for verifying conformance of the gate-level designs with the timing requirements of the individual sub-modules.

16. The method of claim 15, wherein the step of synthesizing gate-level designs is further based on wire loads and input/output loads/drivers.

17. The method of claim 16, wherein the step of verifying conformance of the gate-level designs includes performing dynamic simulations on the gate-level designs.